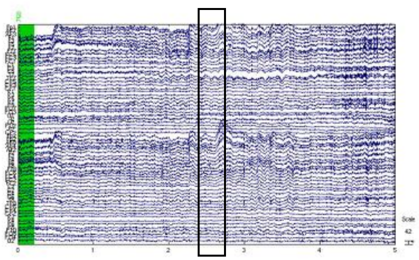
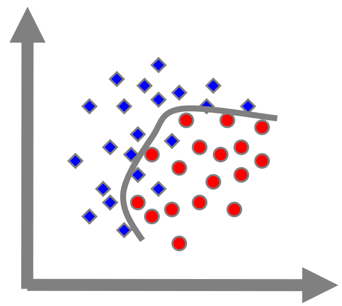
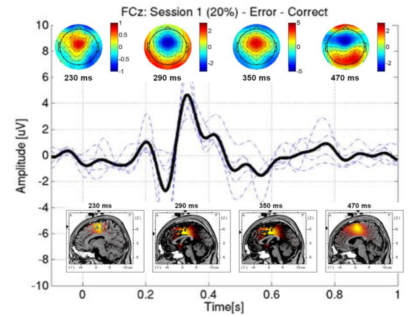






Data analysis and model classification

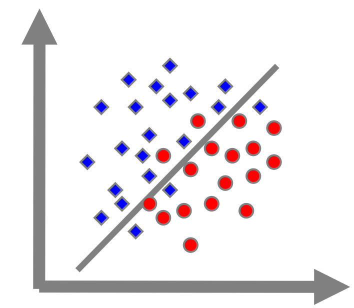
Unsupervised learning

Cross-Validation

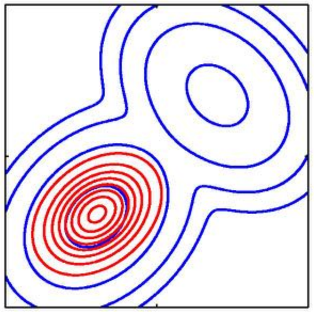
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Types of learning

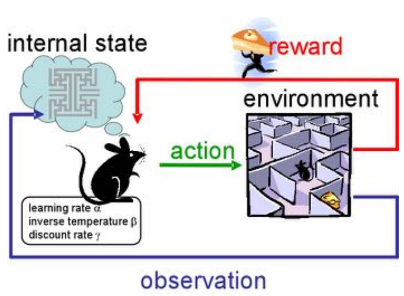
- Supervised learning
  - Learning by examples (inputs and corresponding target values)
  - Minimization of an explicit error function



- Unsupervised learning
  - Model the data distribution without desired target values

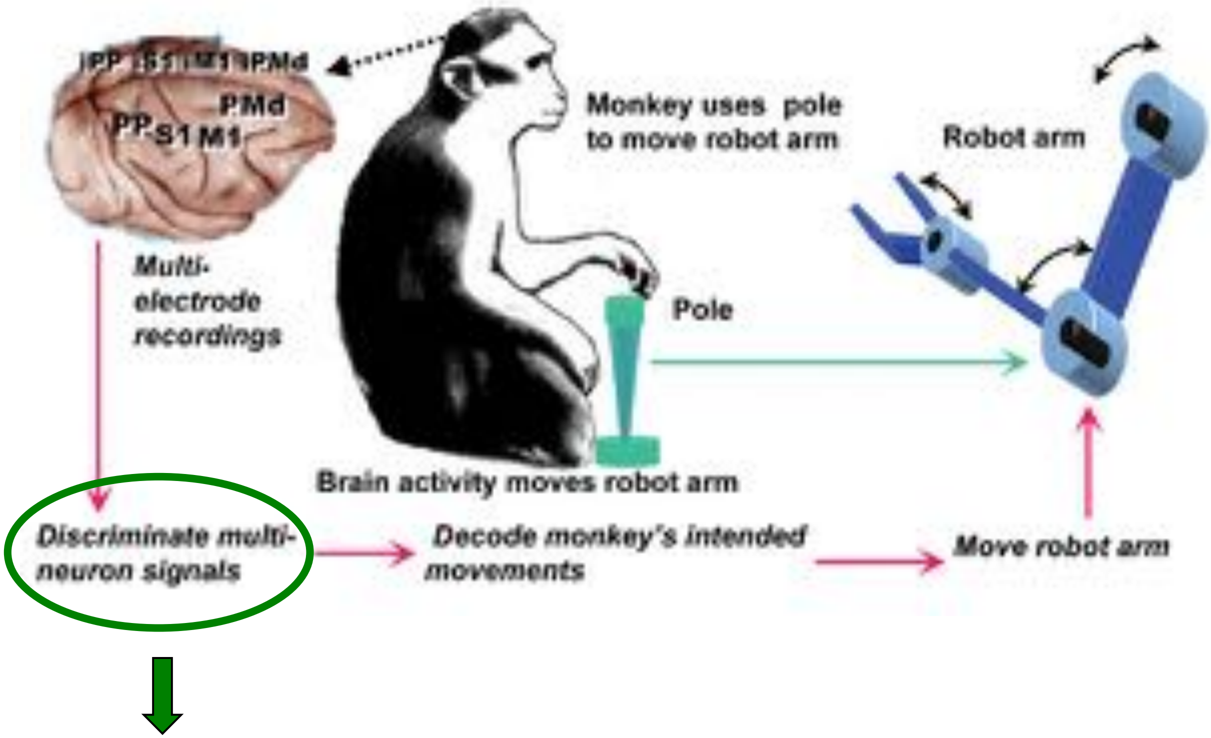


- Semi-supervised learning
  - Information about the performance is provided without explicitly providing target values



Unsupervised Learning

Example: Neuroprosthetics

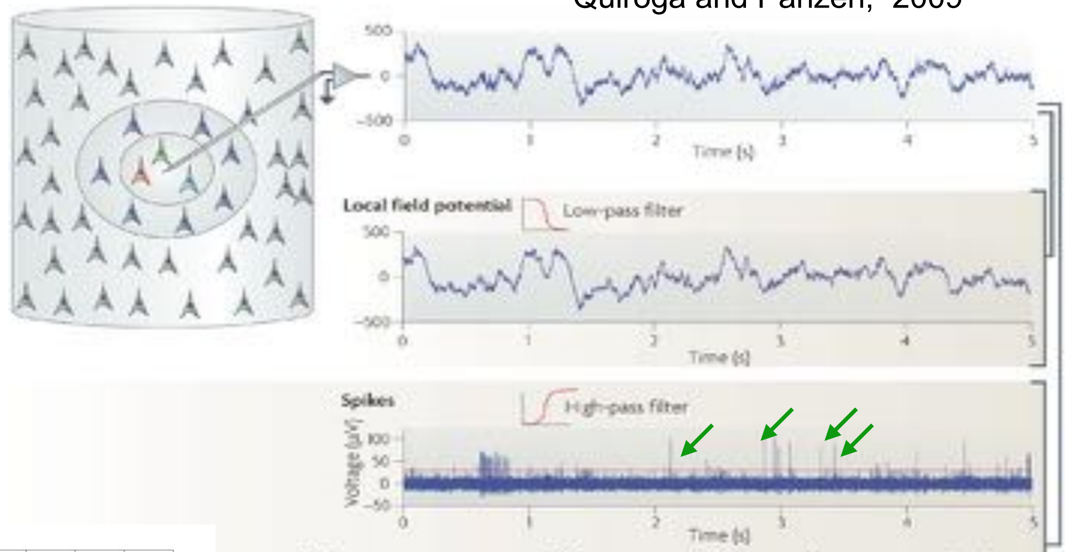


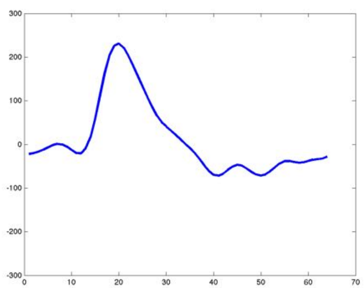
Each implanted electrode records signals from several <u>unknown</u> neurons

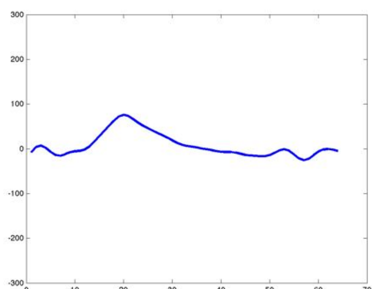
Individual neurons are identified using Unsupervised learning

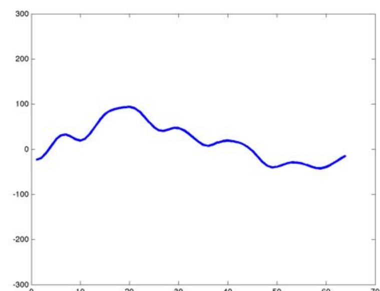
Example: Neuroprosthetics

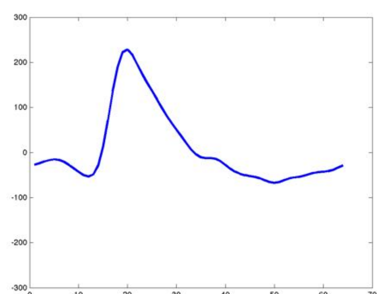
Quiroga and Panzeri, 2009

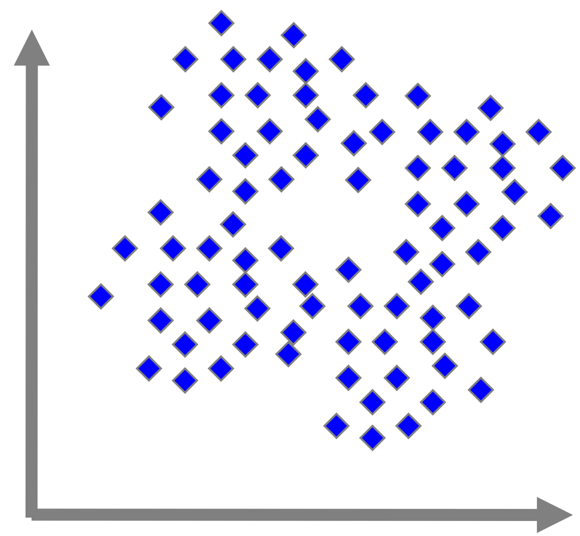






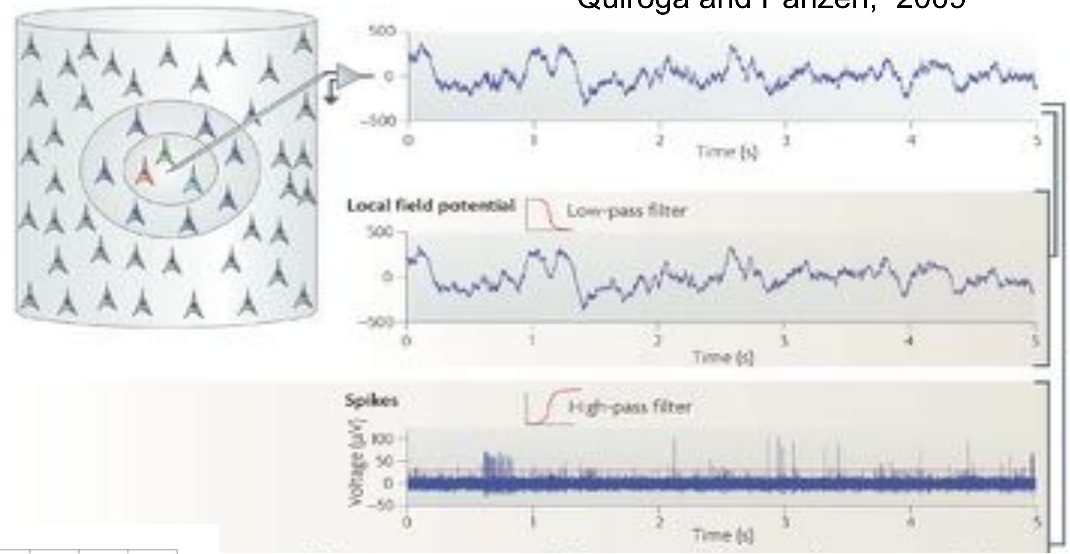


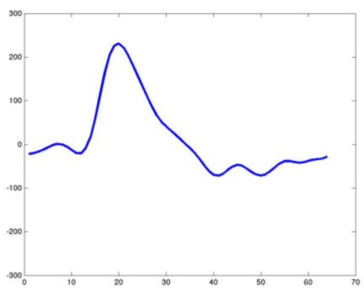


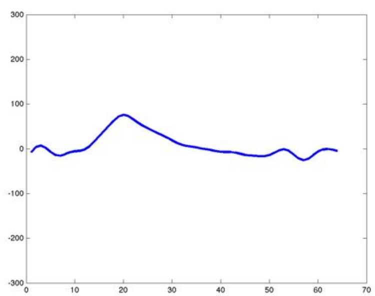


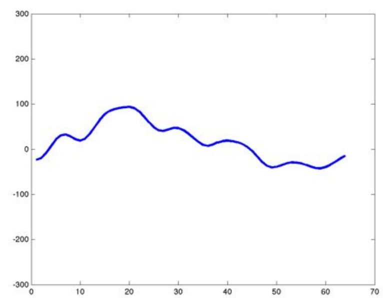
Example: Neuroprosthetics

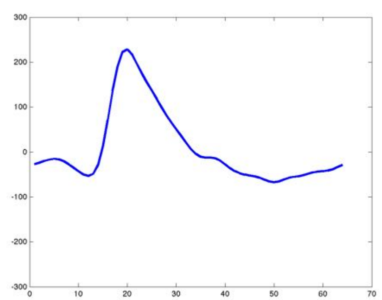
Quiroga and Panzeri, 2009

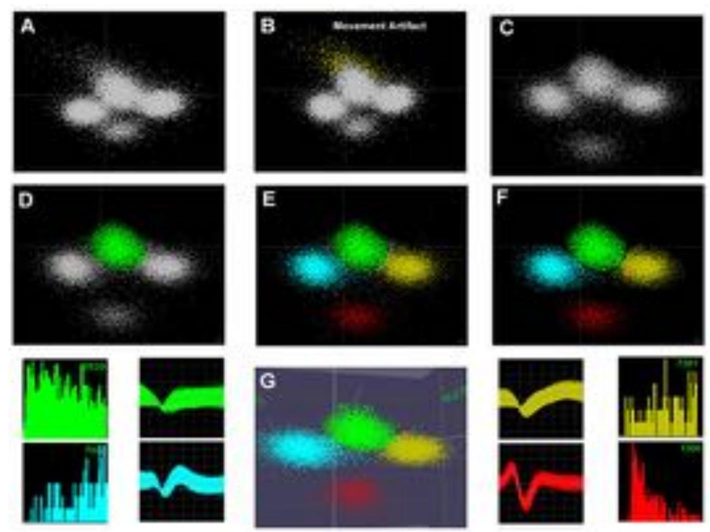










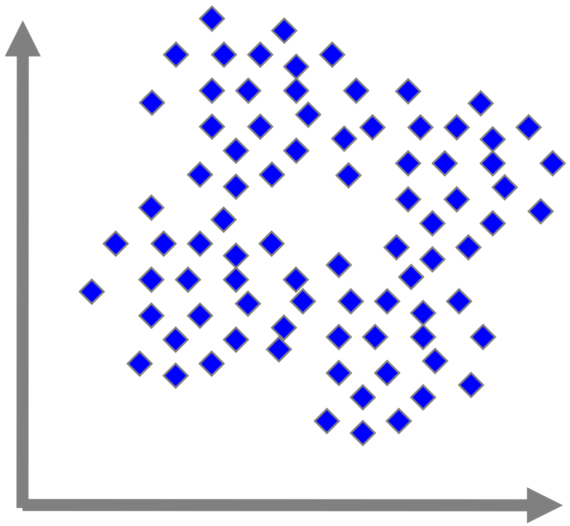


Nicolelis, 2003

Clustering

Characterize the data as an ensemble of groups of data point (clusters)

Cluster: Set of points whose inter-point distances are small compared to points outside the cluster



Clustering: K-Means

Assuming K clusters, we define:

$$\mu_k \equiv \text{Center of cluster k}$$

$$r_{nk} = \begin{cases} 1 & \text{if } \mathbf{x}_n \in \text{cluster k} \\ 0 & \text{otherwise} \end{cases}$$

Goal: minimize objective function

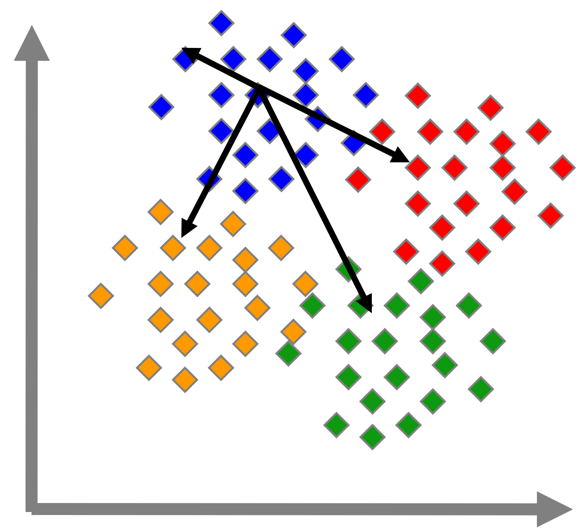
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

Given r_{nk}, J is minimized by $\frac{\partial J}{\partial \mu_k} = 0$

$$2 \sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \mu_k) = 0$$

$$\boxed{\mu_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}}$$

Mean of all points in cluster k



K-Means algorithm

Iterative algorithm

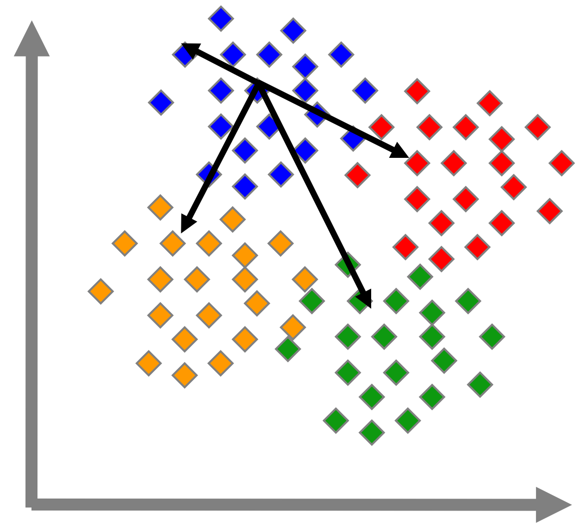
Initialize μ_k
Do
Update r_{nk}:

$$r_{nk} = \begin{cases} 1 & \text{if } k = argmin_j ||\mathbf{x}_n - \mu_j||^2 \\ 0 & \text{otherwise} \end{cases}$$

Update μ_k:

$$\mu_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

Until (no change)

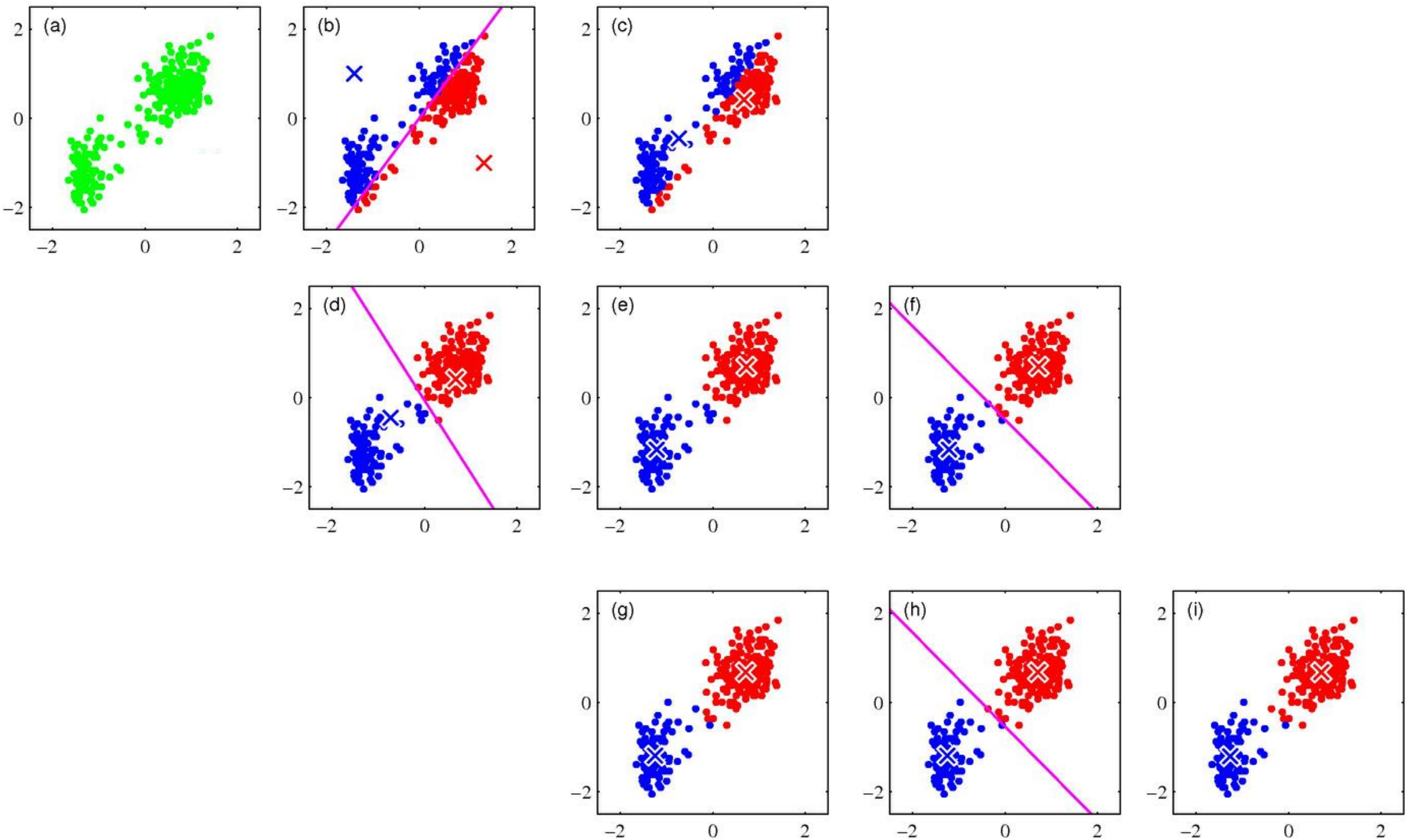


Clustering: K-means

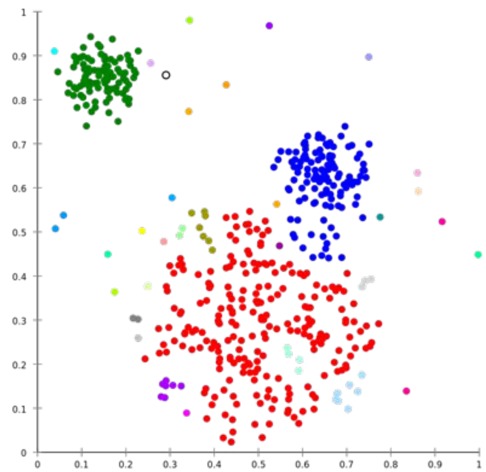
Initialize μ_k

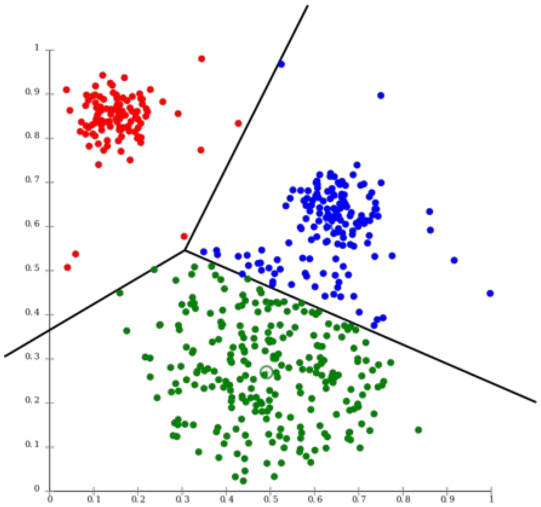
Update r_{nk}:

Update μ_k:



Figs: Bishop et al. Pattern recognition and machine learning, 2006

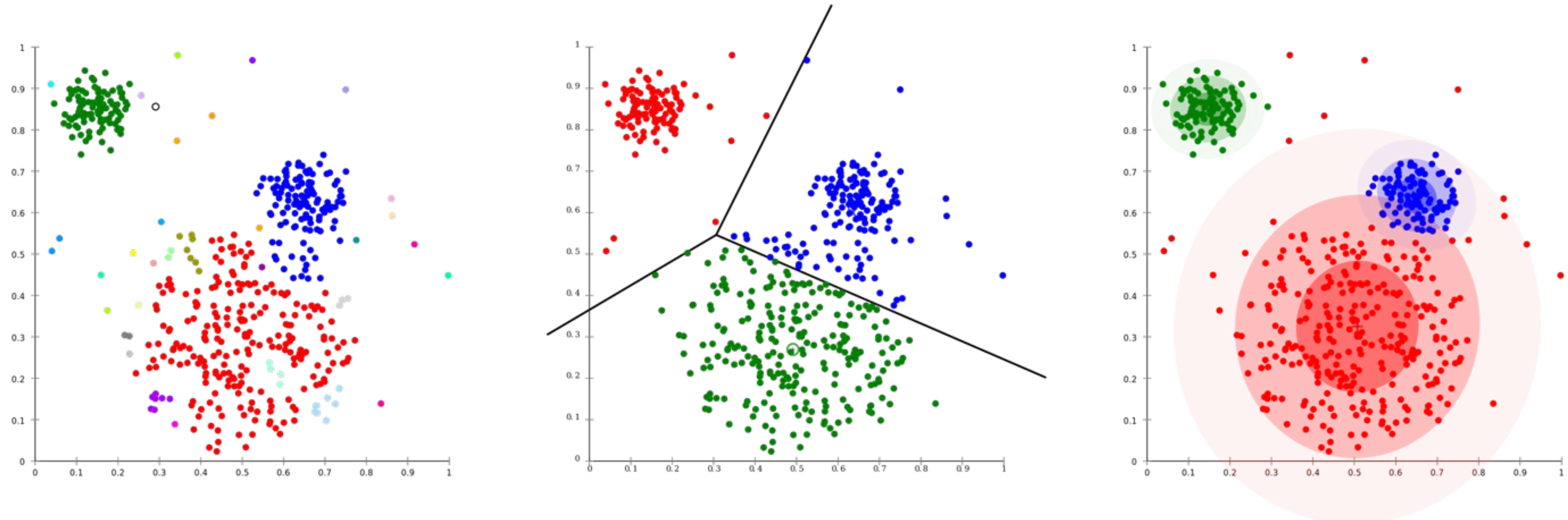




K-means define crisp distinction between clusters

- Feature space is separated into groups of similar size (Voronoi cells)
- Optimization can converge to local minima

Sometimes boundaries between clusters are not necessarily well-defined



K-means define crisp distinction between clusters

- Feature space is separated into groups of similar size (Voronoi cells)
- Optimization can converge to local minima

Sometimes boundaries between clusters are not necessarily well-defined

Mixture densities

Alternatively, the data can be represented as a combination of multiple probability distributions

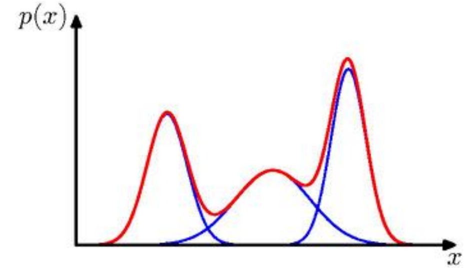
Assuming that data comes from the mixture of K different probability densities

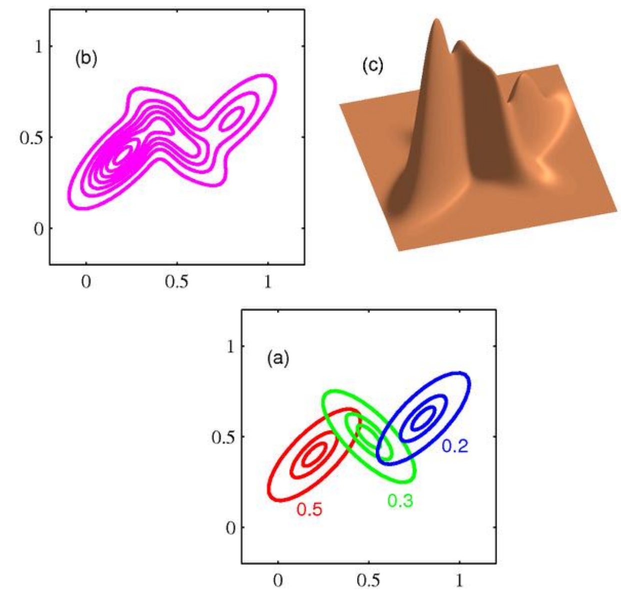
$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k \, p(\mathbf{x}|\theta_k)$$

θ_K : parameters of component density k

π_K : mixing coefficient

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leq \pi_k \leq 1$$

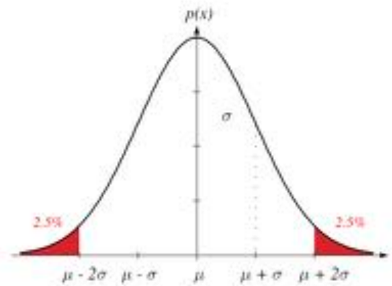




Normal/Gaussian distribution

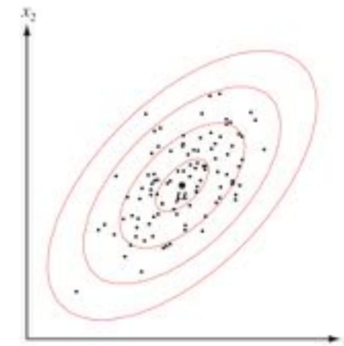
Univariate normal distribution

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

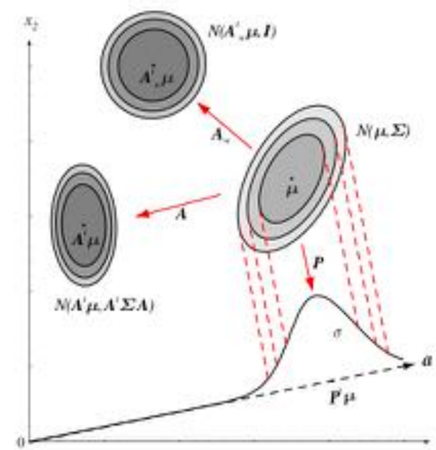


Multivariate normal distribution

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^t\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right]$$



Linear transformation of normal distributions are also normal



Duda et al. Pattern classification, 2nd Ed

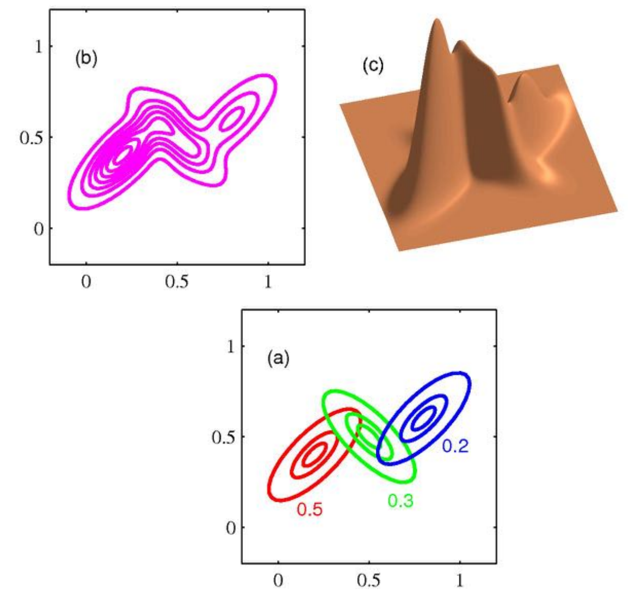
Mixture of Gaussians

Assuming normal distributions and a set of observed samples: $\mathcal{D} = \{\mathbf{x}_1, .., \mathbf{x}_N\}$

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\theta_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)$$

Mixture parameters $\theta = \{\pi, \mu, \Sigma\}$, can be obtained by maximizing the likelihood of the observed samples

$$p(\mathcal{D}|\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)$$



Mixture of Gaussians

Assuming normal distributions and a set of observed samples: $\mathcal{D} = \{\mathbf{x}_1, .., \mathbf{x}_N\}$

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\theta_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)$$

Mixture parameters $\theta = \{\pi, \mu, \Sigma\}$, can be obtained by maximizing the likelihood of the observed samples

$$p(\mathcal{D}|\theta) = \prod_{n=1}^{N} p(\mathbf{x}_n|\theta)$$

Log-likelihood

$$ln\ p(\mathcal{D}|\theta) = \sum_{n=1}^{N} ln\ p(\mathbf{x}_n|\theta)$$

$$ln\ p(\mathcal{D}|\pi, \mu, \mathbf{\Sigma}) = \sum_{n=1}^{N} ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k) \right\}$$

Maximum likelihood

Mixture parameters $\theta = \{\pi, \mu, \Sigma\}$, can be obtained by maximizing the likelihood of the observed samples

$$l = ln\ p(\mathcal{D}|\pi, \mu, \mathbf{\Sigma}) = \sum_{n=1}^{N} ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k) \right\}$$

Setting the derivative of *l* with respect to μ_k equal to zero

$$0 = \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \mathbf{\Sigma}_j)} \mathbf{\Sigma}_k^{-1}(\mathbf{x}_n - \mu_k)$$

$$\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \mathbf{\Sigma}_j)} = p(C_k = 1|\mathbf{x}_n) = \gamma(C_{nk})$$

$$\boxed{\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(C_{nk}) \mathbf{x}_n}$$

$$N_k = \sum_{n=1}^{N} \gamma(C_{nk})$$

Maximum likelihood

We can try to obtain the optimal parameters θ by maximizing the likelihood

$$l = ln\ p(\mathcal{D}|\pi, \mu, \mathbf{\Sigma}) = \sum_{n=1}^{N} ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k) \right\}$$

Setting the derivative of *l* with respect to Σ_k equal to zero

$$\boxed{\mathbf{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(C_{nk})(\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^t}$$

Maximizing *l* with respect to π_k

$$\boxed{\pi_k = \frac{N_k}{N}}$$

Expectation-Maximization

These expressions are not a closed-form solution since they depend on each other.

$$\boxed{\pi_k = \frac{N_k}{N}}$$

$$\boxed{\mu_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(C_{nk})\mathbf{x}_n}$$

$$\boxed{\mathbf{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N}\gamma(C_{nk})(\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^t}$$

Solution:

Compute them iteratively through Expectation-Maximization algorithm

Expectation-Maximization

EM algorithm

Initialize μ Σ π,
Do
E-step: Evaluate posterior probabilities (responsibilities) using current parameters

$$\gamma(C_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \mathbf{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \mathbf{\Sigma}_j)}$$

M-step: Re-estimate the parameters

$$\mu_k^{\mathrm{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(C_{nk}) \mathbf{x}_n$$

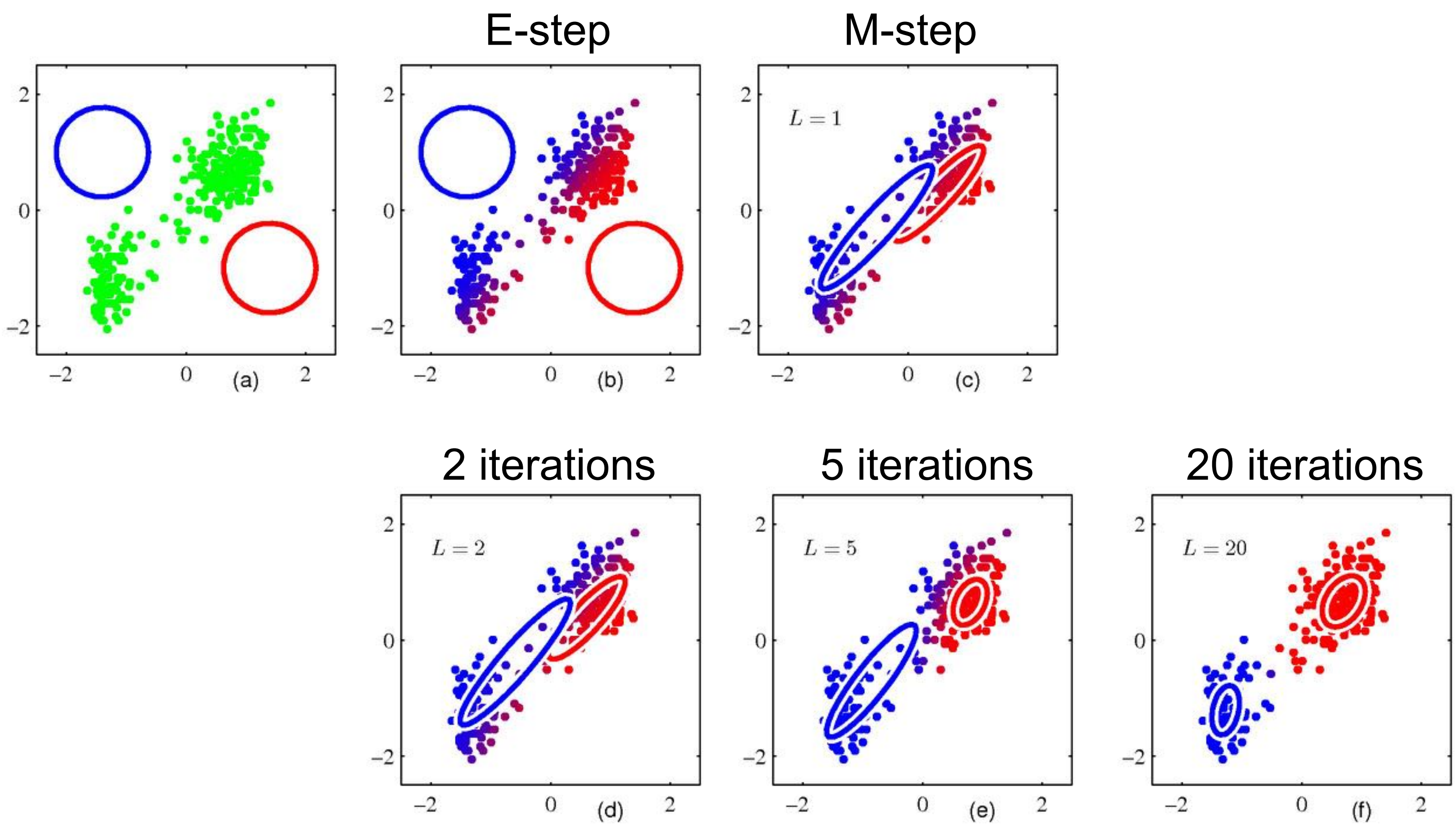
$$\mathbf{\Sigma}_k^{\mathrm{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(C_{nk}) (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^t$$

$$\pi_k^{\mathrm{new}} = \frac{N_k}{N}$$

Evaluate log-likelihood

Until convergence in log-likelihood or parameters

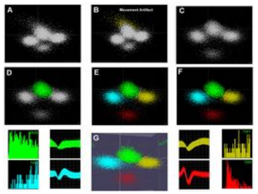
EM Algorithm



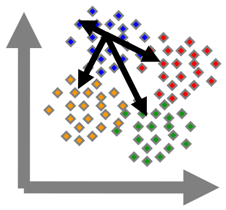
Figs: Bishop et al. Pattern recognition and machine learning, 2006

Summary – Unsupervised learning

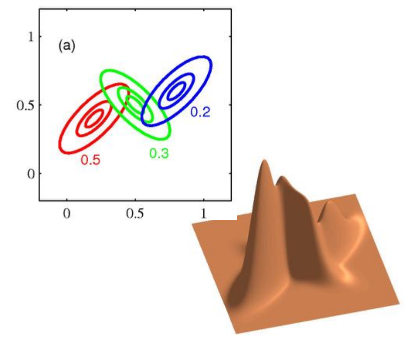
Unsupervised learning is used to process unlabelled data



Data can be characterized by a set of different clusters of data points K-means algorithm

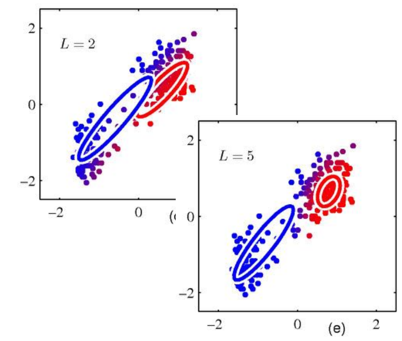


Data can alternatively described as a mixture of density functions



Parameters of the mixture are obtained by maximizing the likelihood of the observed samples

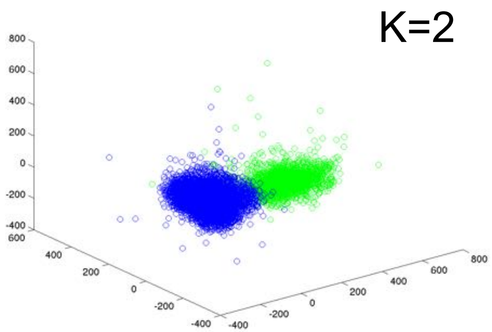
They can be obtained iteratively using the EM algorithm

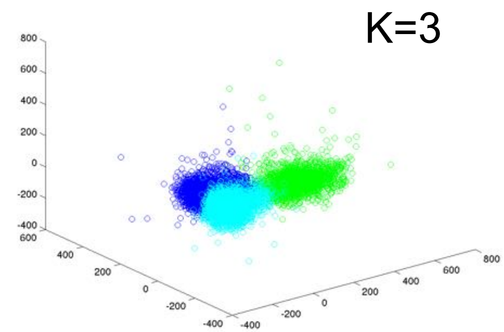


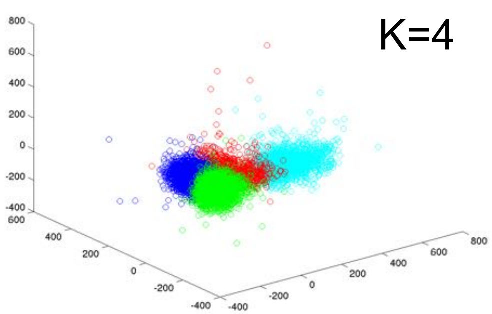
Evaluating the models

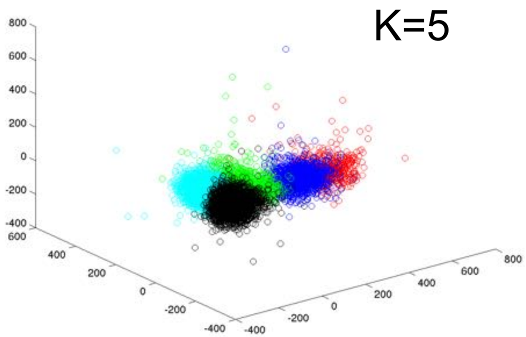
How do I evaluate the **performance** of my model?

How does **performance** depend on the free parameters (e.g., initial conditions, number of clusters)?









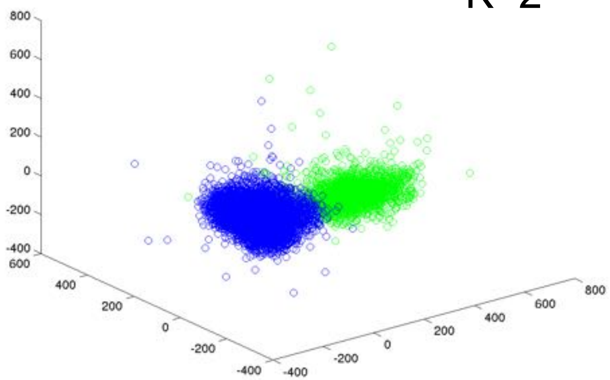
Evaluating the models

How do I evaluate the **performance** of my model?

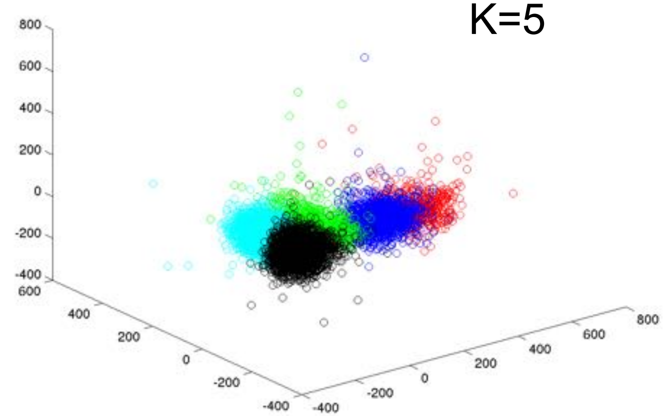
- **Internal** criterion: Based on the same data we used to cluster

- **External** criterion: using external information. e.g., ground truth labels

K=2



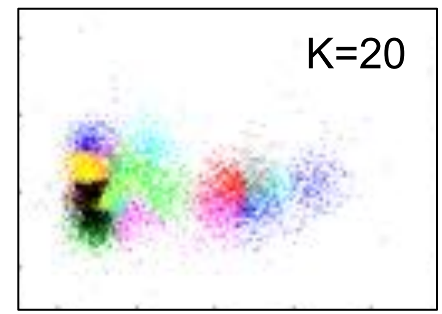
K=5

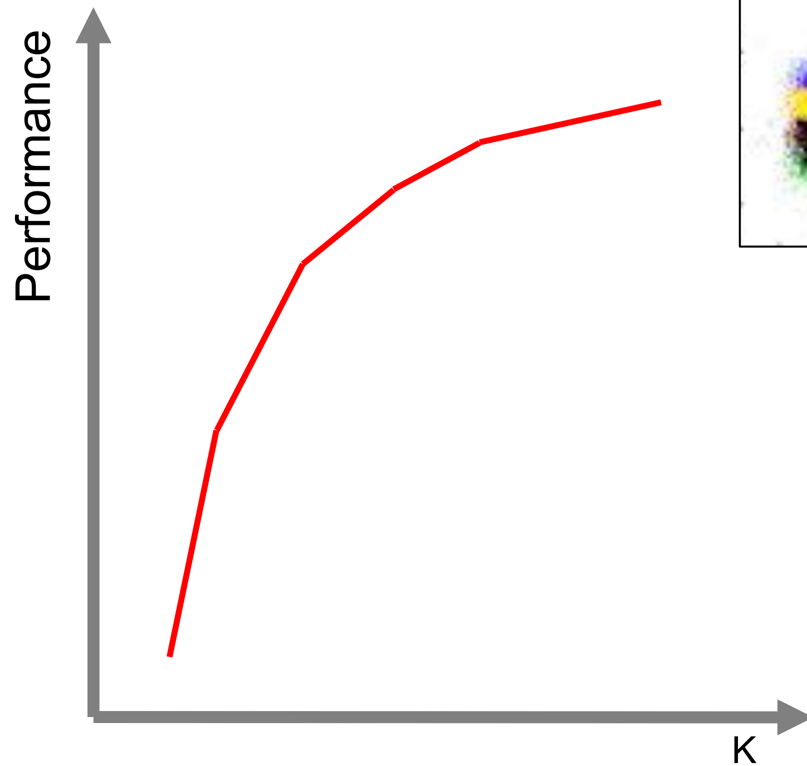


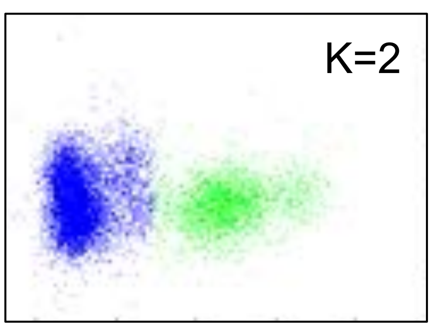
Internal criteria

Does the model reflect my data?

- Explained variance
- Sum of squared distances to centroids







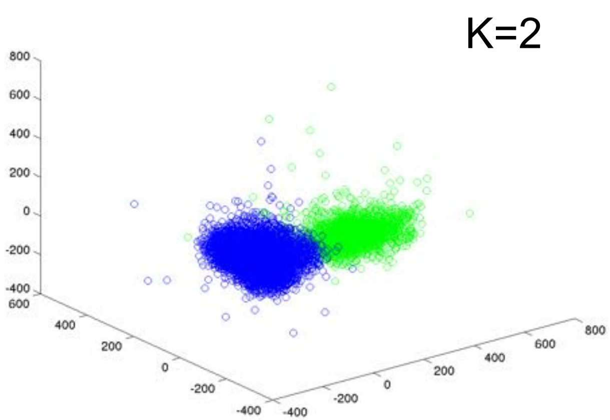
As K >> performance may increase, but also the model's complexity

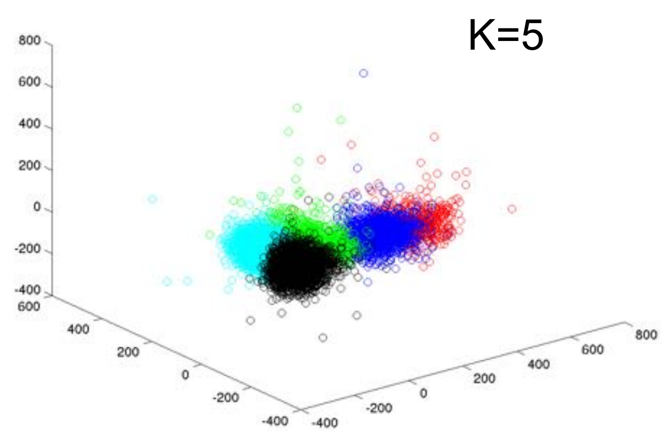
Internal criteria

Performance metrics should reward **high intra-cluster similarity and low inter-cluster similarity**

- **Silhouette**: ratio between average distance of element is the same cluster to average distance to elements in other clusters

- **Dunn index**: Ratio between minimal inter-cluster distance to maximal intra-cluster distance





Internal criteria

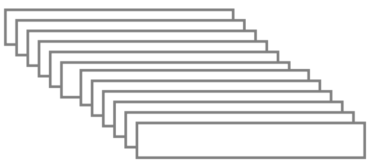
Alternatively we can measure how stable is the outcome of the clustering process

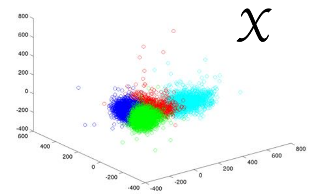
- Do **different initial** conditions always lead to same clusters

Internal criteria

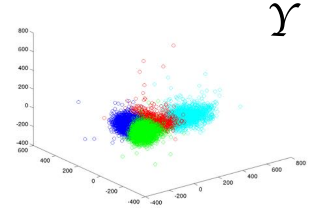
Alternatively we can measure how stable is the outcome of the clustering process

- **Test different partitions** of the available data





$\mathcal{X}$



$\mathcal{Y}$

Given a pair of points (o_i, o_j)

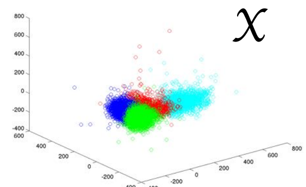
$o_i, o_j \in \mathcal{X}_k$ $o_i, o_j \in \mathcal{Y}_k$	$o_i, o_j \in \mathcal{X}_k$ $o_i \in \mathcal{Y}_{l1}$; $o_j \in \mathcal{Y}_{l2}$
$o_i \in \mathcal{X}_{k1}$; $o_j \in \mathcal{X}_{k2}$ $o_i, o_j \in \mathcal{Y}_l$	$o_i \in \mathcal{X}_{k1}$; $o_j \in \mathcal{X}_{k2}$ $o_i \in \mathcal{Y}_{l1}$; $o_j \in \mathcal{Y}_{l2}$

Internal criteria

Alternatively we can measure how stable is the outcome of the clustering process

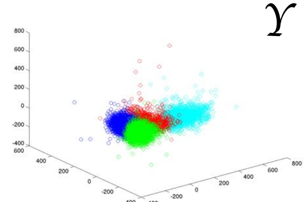
- Test different partitions of the available data

Given a pair of points (o_i, o_j)



$\mathcal{X}$

a	c
d	b



$\mathcal{Y}$

$$\text{Rand index} = \frac{a+b}{a+b+c+d}$$

External criteria

External information (i.e. labels) are used to evaluate performance.

Do samples with the same label get clustered together?

But

If there are labels available, why not use **supervised** learning instead?

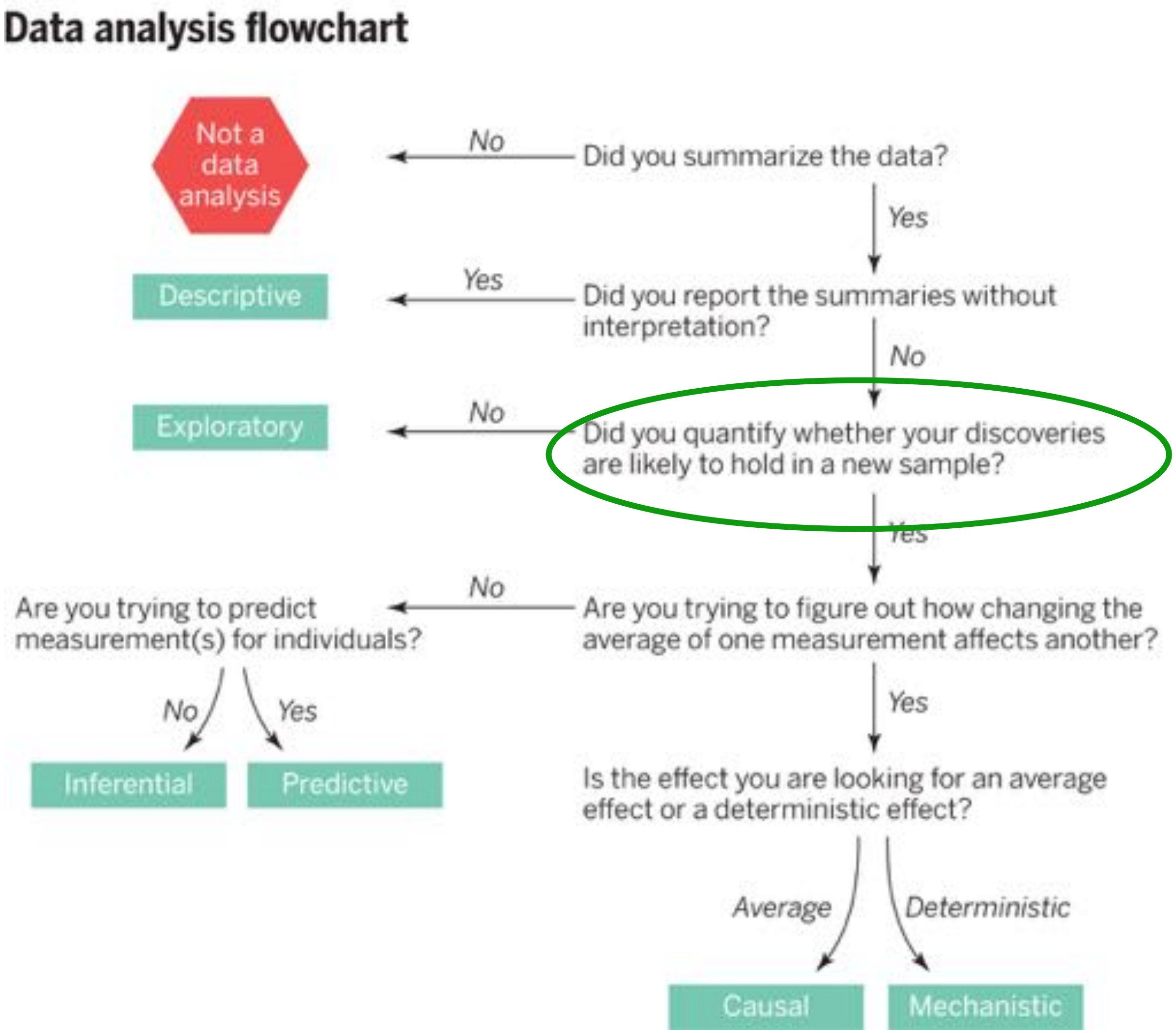
External criteria

External information (i.e. labels) are used to evaluate performance.

Cluster purity: Measure if all samples in a cluster have the same label (class)

Accuracy: Ratio of correctly classified samples to the total number of samples

Data analysis: what for?

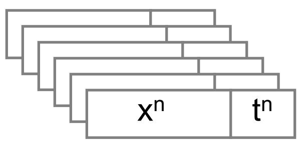


Leek, J. T. & Peng, R. D. Science, 2015

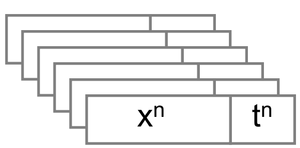
Generalization

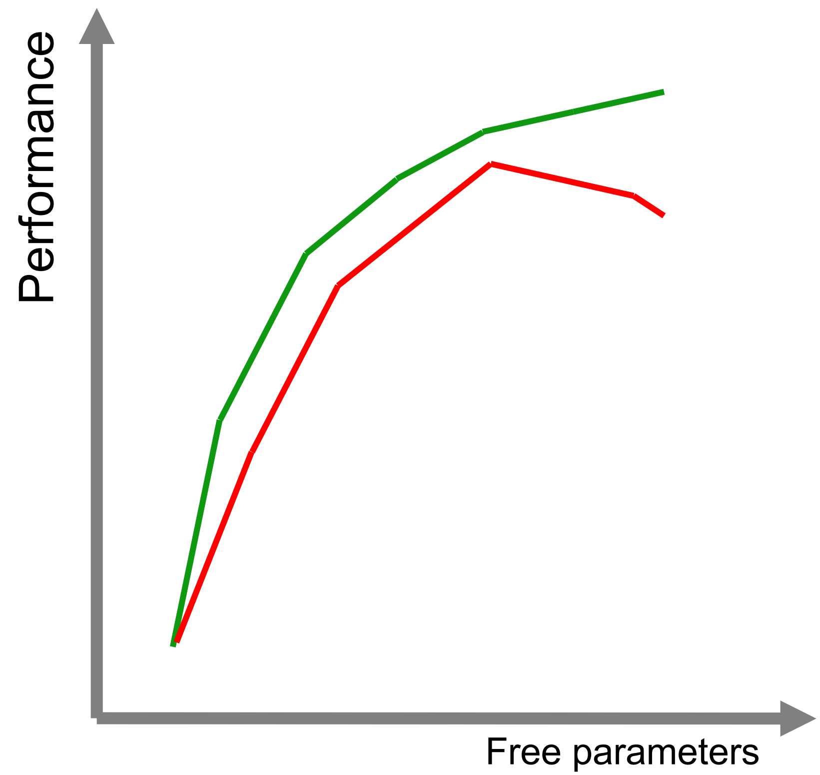
Does the model performance hold for new, unseen inputs

Training set



Test set



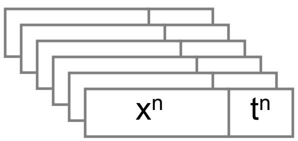


Test set data should not be used to build the model

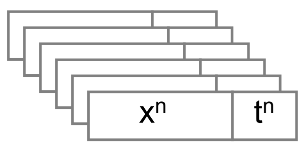
Generalization

Does the model performance hold for new, unseen inputs

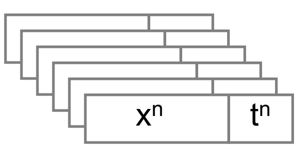
Training set

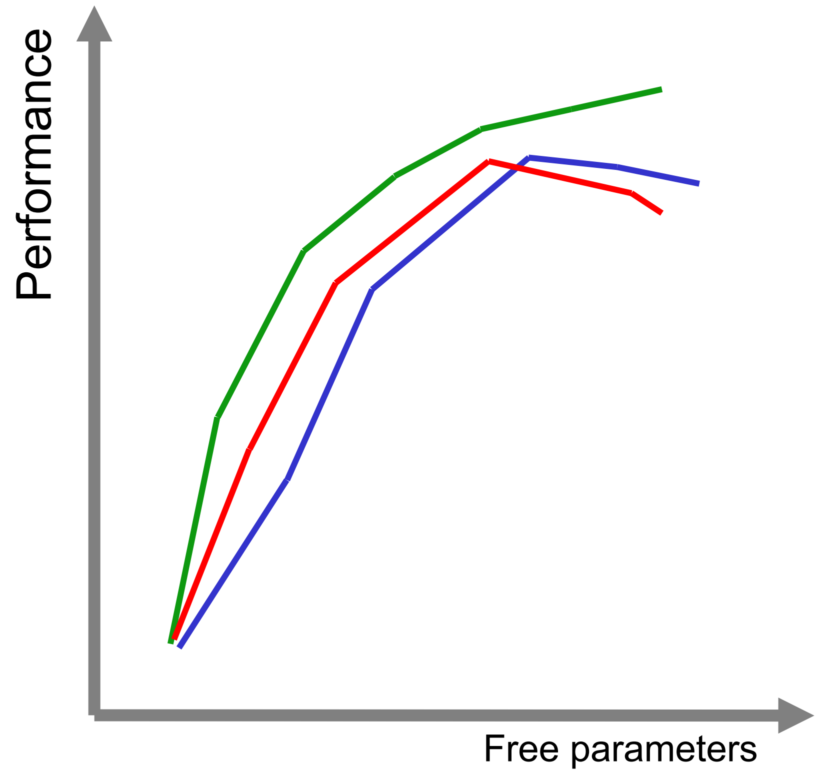


Validation set



Test set





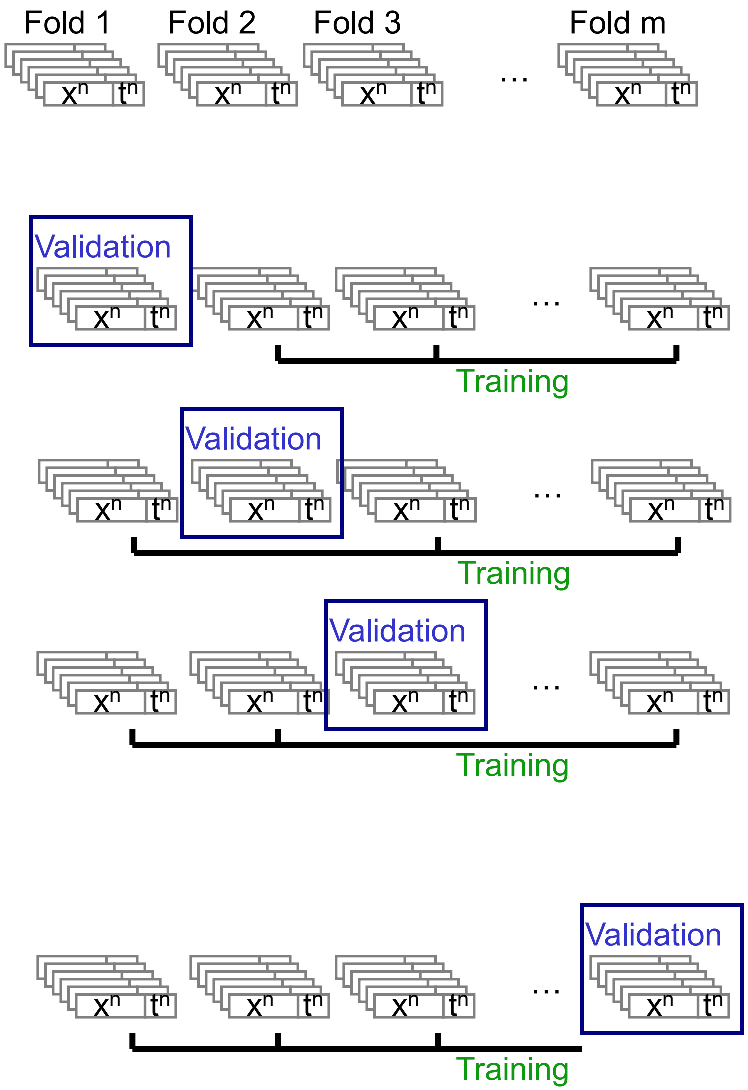
Test set data should not be used to build the model

Therefore, free parameters should not be optimized based on the testing set

m-fold cross-validation

Sometimes amount of data may be not enough to split on the three datasets

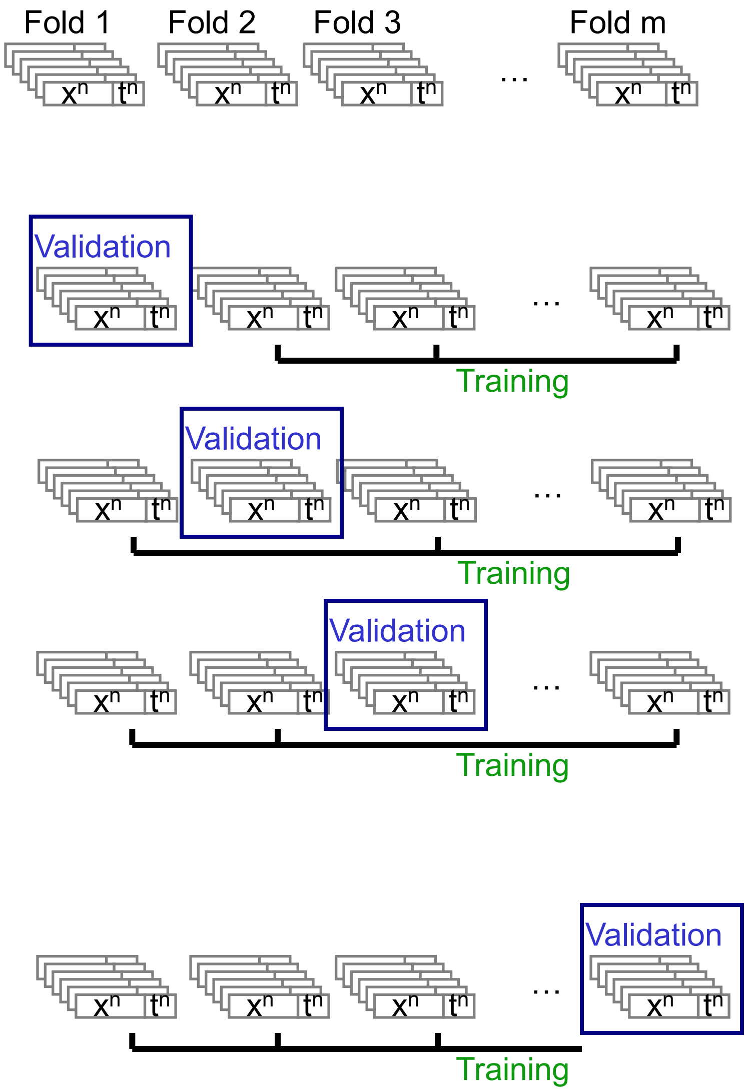
- Split the training set into m disjoint sets (folds) of equal size (N/m)
- Train the classifier m times each time using a different fold as validation
- Provides empirical estimation of the model performance
- Performance is often reported as the average of the m models
- If m=N $\rightarrow$ Leave-one-out



m-fold cross-validation

How to interpret the outcome of the cross-validation process?

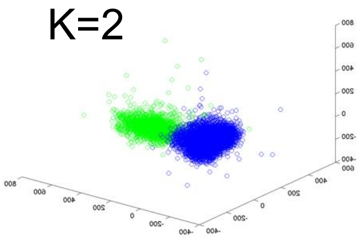
How does CV help in building a system with predictive power for new samples?



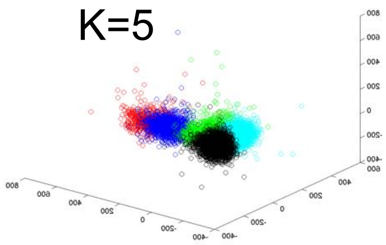
Summary

- Different performance metrics can be used

  There is no silver bullet!

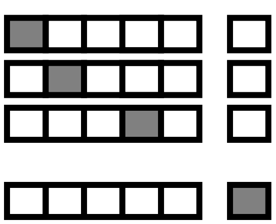


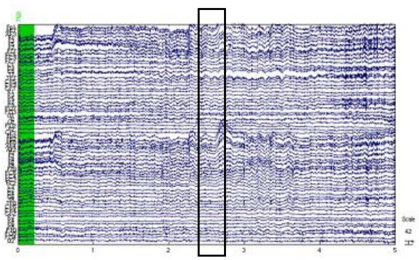
- Criteria can use both internal and external information

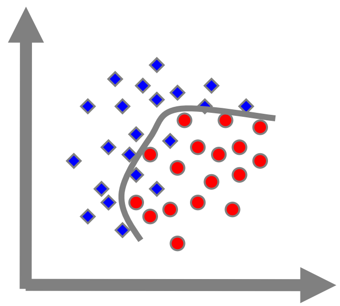


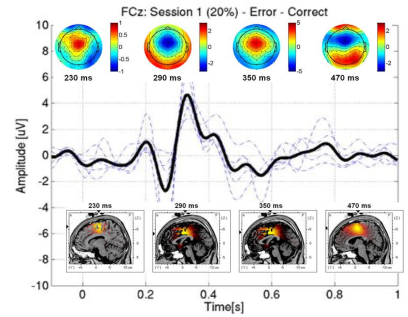
- To assess generalization, the model performance should be assessed on a separate unseen test dataset

- Cross-validation can be used when not enough data is available









Data analysis and model classification

Unsupervised learning

Cross-Validation

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE